Management Summary

It was identified several critical vulnerabilities that pose significant risks to the organisation's IT environment. The current Endpoint Detection and Response (EDR) system is insufficiently effective, which limits the organisation's ability to detect and respond to security threats in a timely manner. Additionally, the ability to upload macro-enabled documents to the WordPress instance presents a serious risk, as these documents can easily be exploited to execute malicious code.

I also discovered that excessive permissions within the network have created opportunities for privilege escalation, allowing unauthorised users to potentially gain domain administrator access. Furthermore, the lack of proper separation between privileged and non-privileged accounts increases the risk of sensitive accounts being compromised. This lack of privileged account segregation makes it easier for attackers to move laterally within the network and escalate their access.

Addressing these vulnerabilities is essential to strengthening the organisation's security posture and reducing the likelihood of a significant security breach. Implementing stronger detection and response measures, restricting the upload of potentially dangerous files, tightening permissions, and ensuring proper account segregation are key steps that should be taken to mitigate these risks.

Predictable Username format

**Description**

The Safebank domain made use of predictable usernames in multiple formats including:

* First initial.surname
* Firstname

When an attacker sends an authentication request to the Kerberos Key Distribution Center (KDC) for a specific username, the KDC's response can inadvertently reveal whether the username is valid or not. Here's how this process works in detail:

How the KDC Responds to Authentication Requests

1. Valid Username:
   * When an authentication request is sent to the KDC for a valid username, such as "sam@example.com," the KDC will typically respond with a KRB5KDC\_ERR\_PREAUTH\_REQUIRED error. This indicates that the username exists, and the KDC is requesting additional pre-authentication data (usually a password).
2. Invalid Username:
   * If the username does not exist, such as "sam123@example.com," the KDC is likely to respond with a KRB5KDC\_ERR\_C\_PRINCIPAL\_UNKNOWN error. This error clearly indicates that the username is not recognised by the KDC.

The difference in responses allows an attacker to easily determine whether a given username is valid, enabling them to enumerate all valid usernames within an organisation.

**Recommendation**

To mitigate this vulnerability, one approach is to avoid using predictable username formats. For instance, instead of using "sam@example.com" as the username, consider using a format that incorporates random elements, such as "sam837@example.com." By adding three random numbers (e.g., "837") to the end of the username, you significantly increase the difficulty for an attacker to guess valid usernames.

Lack of Privilege Separation

**Description**

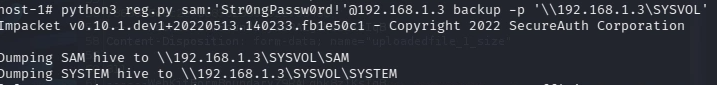
The user account ‘Sam’ had been granted backup operator privileges. This was particularly concerning because backup operators have the ability to back up critical system files, including the Security Accounts Manager (SAM) database on the domain controller.

A screenshot of a computer

Description automatically generated

*Figure 1 - User details*

The SAM database contains hashed passwords for local users, including those with equivalent domain administrator privileges. By exploiting these backup operator privileges, the user could back up the SAM database and potentially extract the hashed passwords, leading to an escalation of privileges to domain administrator level.



*Figure 2 - Backing up SAM database*

A screenshot of a computer

Description automatically generated

*Figure 3 - SAM backed up to SYSVOL share*

This lack of proper privilege separation poses a serious security risk, as it allows a standard user account to perform highly sensitive operations that should be restricted to only the most trusted and secure accounts.

**Recommendation**

To mitigate this risk, it is essential to enforce strict privilege separation within the organisation. Specifically, standard user accounts should not have elevated privileges such as those granted to backup operators. Privileges should be assigned based on the principle of least privilege, where users only have the permissions necessary to perform their job functions.

Additionally, regular audits should be conducted to review and verify that privilege assignments align with organisational security policies. Any accounts that are found to have excessive privileges should be promptly adjusted. For tasks requiring elevated privileges, separate administrative accounts should be used, ensuring that standard user accounts remain limited in their capabilities. This approach will significantly reduce the risk of privilege escalation and protect critical systems from being compromised.

Word Press Self Registration

**Description**

The WordPress instance permitted self-registration, allowing anyone to create an account on the website without prior approval or verification by an administrator. While this may be convenient for users, it also opens up the system to potential abuse by malicious actors, this was particularly true in this case as the site had a feature for uploading files.

A screenshot of a computer screen

Description automatically generated

Figure 4 - File Upload

As an authorised user, it may also aid in exploiting vulnerabilities in plugins and themes that often need authentication.

**Recommendation**

To mitigate the risks associated with WordPress self-registration, it is recommended to disable the self-registration feature unless absolutely necessary. If user registration is required, implement additional security measures such as email verification, CAPTCHA challenges, and manual approval by an administrator to ensure that only legitimate users gain access. Additionally, regularly review and audit user accounts to identify and remove any suspicious or inactive accounts. These steps will help protect the WordPress instance from unauthorised access and potential exploitation.

Insufficient EDR Solution

**Description**

The security assessment revealed that the current Endpoint Detection and Response (EDR) solution in place is insufficient in detecting and preventing malicious activities. This inadequacy was demonstrated by the ability to execute malicious code through a macro-enabled .doc file. Attackers can embed harmful macros in Word documents, which, when opened by a user, can execute malicious code on the victim's machine. The EDR solution failed to detect and block this activity, allowing the attacker to bypass security controls and potentially gain further access to the network.

This weakness exposes the organisation to a variety of risks, including the execution of ransomware, data theft, and further compromise of systems within the network. The ability of the macro-enabled document to execute code without being detected highlights the need for a more robust security solution.

**Recommendation**

To address this vulnerability, it is crucial to upgrade or supplement the existing EDR solution with one that offers more advanced threat detection capabilities, particularly for detecting and blocking macro-based attacks. The new or enhanced EDR solution should include features such as behavioral analysis, heuristic scanning, and real-time threat intelligence updates to identify and prevent the execution of malicious code from macro-enabled documents.

Additionally, consider implementing stricter policies on the use of macro-enabled documents within the organisation. This may include disabling macros by default, only allowing macros from trusted locations, or using document management systems that scan and sanitize files before they are opened. Regularly training employees on the risks associated with macro-enabled documents and how to recognise potentially malicious files can also help reduce the likelihood of successful attacks.